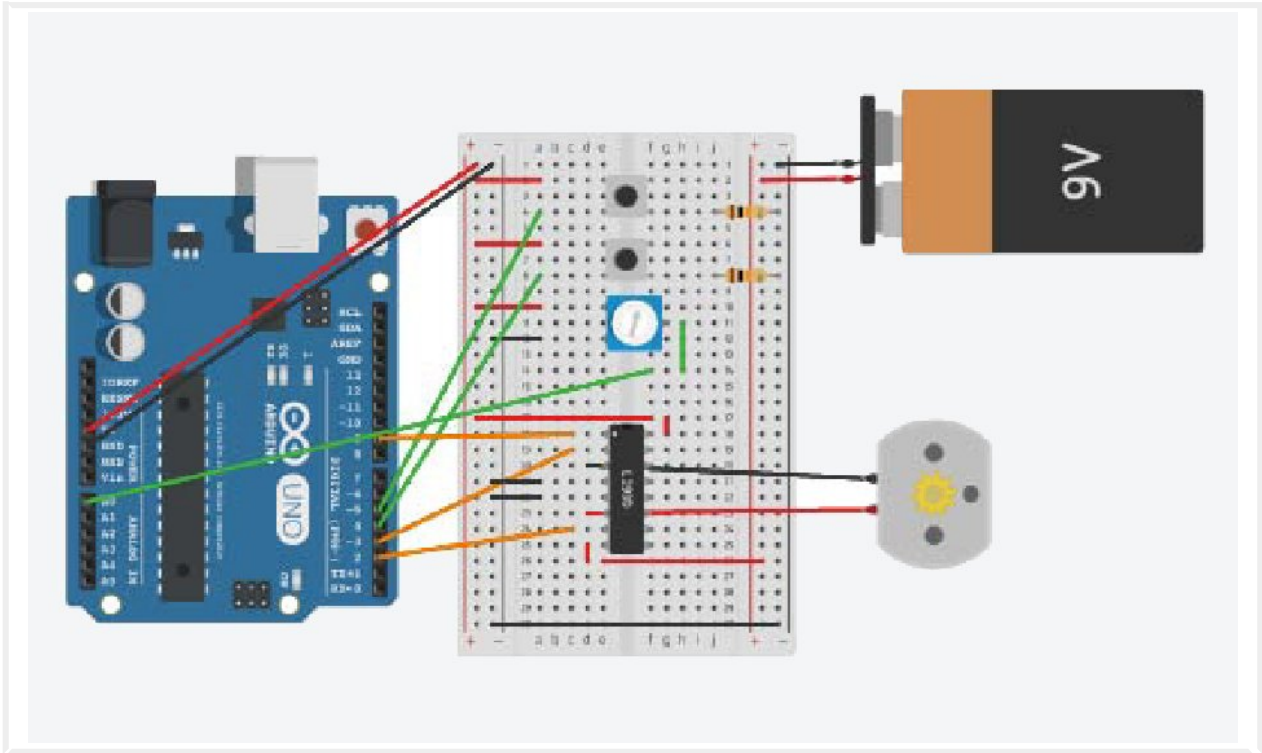


Moteur DC - L293D

*(Gestion de la rotation d'un moteur à courant continu
dans les 2 sens et à vitesse variable)*



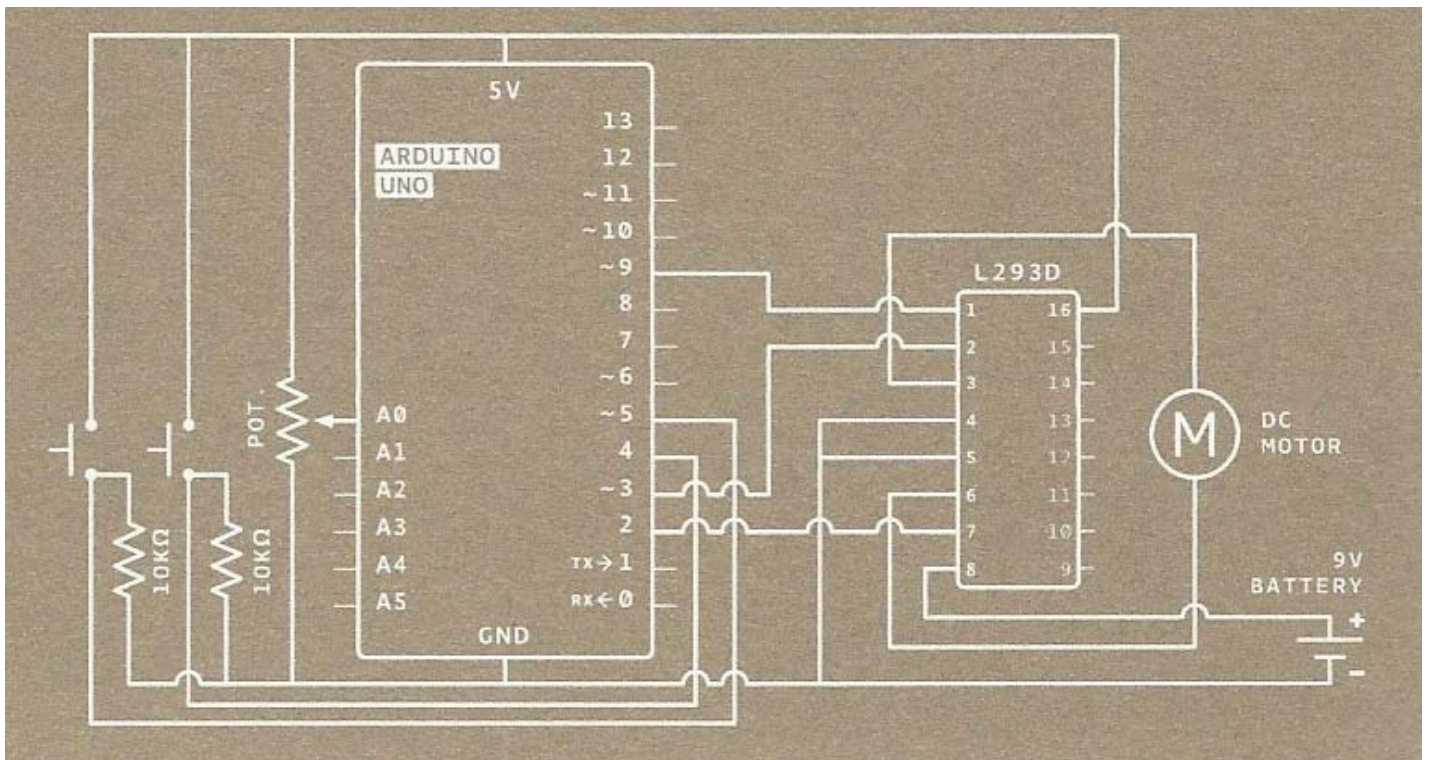
. Liste des composants

- . 1 Moteur DC
- . 2 boutons poussoir
- . 1 potentiomètre de 10 k Ω
- . 2 résistances de 10 k Ω
- . 1 CI L293D
- . 1 Pile 9 V
- . Fils de connexion

. Objectif

L'objectif de cette activité est de faire tourner un moteur DC dans les deux sens à vitesse de rotation variable.

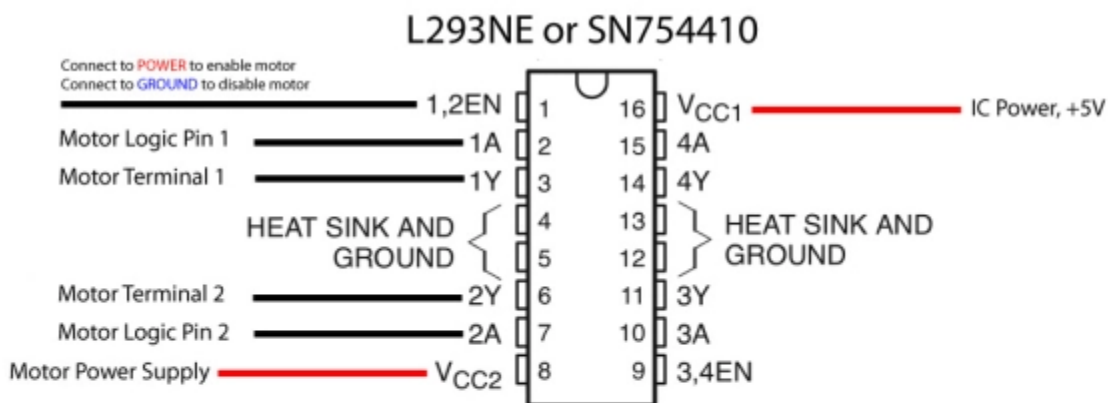
Pour cela on utilise un pont en H, le CI L293D, selon le montage suivant :



Caractéristiques techniques du L293D

Le composant L293D est un pont de puissance composé de plusieurs transistors et relais qui permet d'activer la rotation d'un moteur.

Le L293D est un double pont-H. Il est possible de l'utiliser pour commander deux moteurs distincts dans les deux sens grâce à ses 4 canaux.



EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

- Courant Max Régime continu : 600mA (x2)
- Courant de pointe Max < 2ms : 1200mA
- VS Max Alim moteur : 36v
- VSS Max Alim logique : 7v
- Nombre de broches : 16 DIP
- Perte de tension : 1.3v

- . **1,2EN** permet d'envoyer (ou pas) la tension sur les sorties du moteur via **1Y** et **2Y** et commande l'activation/désactivation du premier Pont-H. Si **1,2EN = GND**, le pont-H est déconnecté et le moteur ne fonctionne pas. Si **1,2EN = VSS**, le pont-H est connecté aux sorties et le moteur peut alors fonctionner dans un sens ou dans l'autre si les tensions appliquées sur **1A** et **2A** sont différentes (cf. tableau ci-dessus).
- . **1A** et **2A** sont les broches de commande du Pont-H **1Y / 2Y**. Ils sont directement branchés à l'Arduino pour commander le sens du courant entre **1Y** et **2Y**.
- . **1Y** et **2Y** sont branchés directement sur le moteur.
- . **GROUND** doit être raccordé à la masse de la source d'alimentation de puissance **VCC₂**(moteur) et à la masse de la source d'alimentation de l'Arduino.
- . **VCC₂** : Alimentation de puissance des moteurs.
- . **3,4EN** commande l'activation du second pont-H constitué de **3Y / 4Y**
- . **3A** et **4A** sont les broches de commande du Pont-H **3Y / 4Y**. Ils sont directement branchés à l'Arduino pour commander le sens du courant entre **3Y** et **4Y**.
- . **3Y** et **4Y** sont branchés directement sur le moteur.

. Le programme

Voici le code de l'activité :

```
Moteur_DC_-_L293D

const int controlPin1 = 2;
const int controlPin2 = 3;
const int enablePin = 9;
const int directionSwitchPin = 4;
const int onOffSwitchStateSwitchPin = 5;
const int potPin = A0;

int onOffSwitchState = 0;
int previousOnOffSwitchState = 0;
int directionSwitchState = 0;
int previousDirectionSwitchState = 0;

int motorEnabled = 0;
int motorSpeed = 0;
int motorDirection = 1;

void setup() {

    pinMode(directionSwitchPin, INPUT);
    pinMode(onOffSwitchStateSwitchPin, INPUT);
    pinMode(controlPin1, OUTPUT);
    pinMode(controlPin2, OUTPUT);
    pinMode(enablePin, OUTPUT);

    digitalWrite(enablePin, LOW);
}
```

```

void loop() {

    onOffSwitchState = digitalRead(onOffSwitchStateSwitchPin);
    delay(1);

    directionSwitchState = digitalRead(directionSwitchPin);

    motorSpeed = analogRead(potPin)/4;

    if(onOffSwitchState != previousOnOffSwitchState){

        if(onOffSwitchState == HIGH){
            motorEnabled = !motorEnabled;
        }
    }

    if (directionSwitchState != previousDirectionSwitchState) {

        if (directionSwitchState == HIGH) {
            motorDirection = !motorDirection;
        }
    }

    if (motorDirection == 1) {
        digitalWrite(controlPin1, HIGH);
        digitalWrite(controlPin2, LOW);
    }
    else {
        digitalWrite(controlPin1, LOW);
        digitalWrite(controlPin2, HIGH);
    }

    if (motorEnabled == 1) {

        analogWrite(enablePin, motorSpeed);
    }
    else {
        analogWrite(enablePin, 0);
    }

    previousDirectionSwitchState = directionSwitchState;

    previousOnOffSwitchState = onOffSwitchState;
}

```

Principe de fonctionnement :

En pressant le premier bouton-poussoir (Bouton On/OFF), le moteur commence à tourner. La vitesse de rotation est réglable grâce au potentiomètre. En pressant une nouvelle fois le premier bouton-poussoir, le moteur s'arrête.

Le sens de rotation du moteur est défini par une pression sur le deuxième bouton-poussoir quand le moteur fonctionne.

Déroulement du programme :

– 1. Déclaration des constantes et variables :

- . const int controlPin1 = 2 (constante nombre entier correspondant à la broche de l'Arduino reliée à la broche 7 du L293D)
- . const int controlPin2 = 3 (constante nombre entier correspondant à la broche de l'Arduino reliée à la broche 2 du L293D)
- . const int enablePin = 9 (constante nombre entier correspondant à la broche de l'Arduino reliée à la broche 1 du L293D pour activer la rotation du moteur)
- . const int directionSwitchPin = 4 (constante nombre entier correspondant à la broche du bouton poussoir pour le sens de rotation du moteur)
- . const int onOffSwitchStateSwitchPin = 5 (constante nombre entier correspondant à la broche du bouton poussoir pour la mise en route du moteur)
- . const int potPin = A0 (constante nombre entier correspondant à la broche du potentiomètre pour le réglage de la vitesse de rotation)
- . int onOffSwitchState = 0 (variable nombre entier pour stocker la valeur du potentiel de la broche du bouton poussoir pour la mise en route du moteur)
- . int previousOnOffSwitchState = 0 (variable nombre entier pour stocker l'ancienne valeur du potentiel de la broche du bouton poussoir pour la mise en route du moteur)
- . int directionSwitchState = 0 (variable nombre entier pour stocker la valeur du potentiel de la broche du bouton poussoir pour le sens de rotation du moteur)
- . int previousDirectionSwitchState = 0 (variable nombre entier pour stocker l'ancienne valeur du potentiel de la broche du bouton poussoir pour le sens de rotation du moteur)
- . int motorEnabled = 0 (variable nombre entier indiquant si le moteur fonctionne)
- . int motorSpeed = 0 (variable nombre entier pour la vitesse de rotation)
- . int motorDirection = 1 (variable nombre entier pour le sens de rotation)

– 2. Initialisation des entrées et sorties :

- . **Initialisation de la broche du bouton poussoir pour la mise en route du moteur en entrée,**
- . **Initialisation de la broche du bouton poussoir pour le sens de rotation du moteur en entrée,**
- . **Initialisation des broches de contrôle du moteur (broches « controlPin1 », « controlPin2 » et « enablePin ») en sortie,**
- . **La broche « enablePin » pour l'activation de la rotation du moteur est mise à un niveau bas.**

– 3. Fonction principale en boucle :

- > Lecture de la valeur de la broche du bouton poussoir pour la mise route du moteur,
- > Lecture de la valeur de la broche du bouton poussoir pour le sens de rotation,
- > Réglage de la vitesse de rotation du moteur après lecture de la valeur de la broche du potentiomètre,
- > Mise en rotation du moteur ou arrêt du moteur si changement de la valeur du bouton poussoir correspondant,
- > Changement du sens de rotation si appui sur le bouton poussoir correspondant.